Homework 5

Christophe Hunt

March 4, 2017

Contents

1 Problem Set 1 **1**
1.1 Write R Markdown script to compute A^TA and A^Tb . . . 1
1.2 Solve for $\hat{x}$ in R using the above computed matrices . . . 2
1.3 What is the squared error of this solution? . . . 2
1.4 Find the exact solution with p instead of b . . . 2

2 Problem Set 2 **4**
2.1 Solve for $\hat{x}$ in R using the above computed matrices . . . 5
2.2 What is the squared error of this solution? . . . 5

1 Problem Set 1

Consider the unsolvable system Ax = b as given below:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \\ 8 \\ 20 \end{bmatrix}$$

1.1 Write R Markdown script to compute A^TA and A^Tb

```
A <- matrix(c(1,1,1,1,0,1,3,4), ncol = 2)
b <- matrix(c(0,8,8,20))

ATA <- t(A) %*% A
ATb <- t(A) %*% b

results <- list("ATA" = ATA, "ATb" = ATb)
results
```

```
## $ATA
##      [,1] [,2]
## [1,]    4    8
## [2,]    8   26
##
## $ATb
##      [,1]
## [1,]   36
## [2,]  112
```

1.2 Solve for $\hat{x}$ in R using the above computed matrices

```
x <- solve(ATA) %*% ATb
x
```

```
##      [,1]
## [1,]    1
## [2,]    4
```

1.3 What is the squared error of this solution?

```
p <- A %*% x
#b = p + e or e = p - b which we can substitute in our given values.
e <- p - b
# we then sum the square of errors.
e2 <- sum(e^2)
e2
```

```
## [1] 44
```

1.4 Find the exact solution with p instead of b

```
options(scipen = 999)
p   <- matrix(c(1,5,13,17))
ATp <- t(A) %*% p
xp  <- solve(ATA) %*% ATp
p2  <- A %*% xp
e   <- p2-p
e
```

```
##                          [,1]
## [1,] 0.00000000000000000000000
## [2,] 0.00000000000000008881784
## [3,] 0.00000000000000035527137
## [4,] 0.00000000000000035527137
```

Essentially, the error vector e is = 0.

```
e2p <- sum(e^2)
e2p
```

```
## [1] 0.0000000000000000000000000000002603241
```

Show that the error $e = b - p = [-1; 3; -5; 3]$.

```
b - p
```

```
##      [,1]
## [1,]   -1
## [2,]    3
## [3,]   -5
## [4,]    3
```

Show that the error e is orthogonal to p and to each of the columns of A.

As per the week 5 handout - We know that when two vectors are orthogonal, their dot product is zero.

```r
e*p
```

```
##                          [,1]
## [1,] 0.000000000000000000000
## [2,] 0.000000000000004440892
## [3,] 0.000000000000046185278
## [4,] 0.000000000000060396133
```

```r
sum(e*A[,1])
```

```
## [1] 0.000000000000007993606
```

2 Problem Set 2

Write an R markdown script that takes in the auto-mpg data, extracts an A matrix from the first 4 columns and b vector from the fifth (mpg) column.

Apparently, an added column of 1 is necessary to obtain an intercept.

```
x <- as.matrix(read.table("https://raw.githubusercontent.com/ChristopheHunt/MSDA---Coursework/master/Da

A <- as.matrix(cbind(x[,1:4],1))
b <- as.matrix(x[,5])
```

Using the least squares approach, your code should compute the best fitting solution

```
ATA <- t(A) %*% A
ATb <- t(A) %*% b
results <- list("ATA" = ATA, "ATb" = ATb)
results
```

```
## $ATA
##              V1          V2         V3         V4
## V1   19097634.2   9374647.0  259345480  1123011.9   76209.5
## V2    9374647.0   4857524.0  132989885   607832.3   40952.0
## V3  259345480.0 132989885.0 3757575489 17758103.6 1167213.0
## V4    1123011.9    607832.3   17758104    97656.9    6092.2
##         76209.5     40952.0    1167213     6092.2     392.0
##
## $ATb
##           [,1]
## V1   1529685.9
## V2    868718.8
## V3  25209061.4
## V4    146401.4
##         9190.8
```

2.1 Solve for $\hat{x}$ in R using the above computed matrices

```
x <- solve(ATA) %*% ATb
x
```

```
##              [,1]
## V1 -0.006000871
## V2 -0.043607731
## V3 -0.005280508
## V4 -0.023147999
##    45.251139699
```

The least squares model using this method is:

$$mpg = -0.006 * displacement + -0.04361 * horsepower + -0.00528 * weight + -0.02315 * acceleration + 45.25114$$

Finally, calculate the fitting error between the predicted mpg of youur model and actual mpg.

2.2 What is the squared error of this solution?

```
p <- A %*% x
#b = p + e or e = p - b which we can substitute in our given values.
e <- p - b
# we then sum the square of errors.
e2 <- sum(e^2)
e2
```

```
## [1] 6979.413
```